COMPARISON IN FUNCTIONALITY BETWEEN A CLOSED AND TWO OPEN SOURCE DISTRIBUTIONS IN A ROUTER

IT604G, VT2016

Jacob Carlsson

a13jacca@student.his.se

2016-07-21

Final Final Report

Supervisor: Jianguo Ding
Examiner: Jonas Mellin

Report

School of Informatics
University of Skövde

Abstract

With open source router firmware being used for various tasks that would be hard to achieve for the standard closed source router firmware, it is important to compare the two in terms of performance. This study aimed to study the differences in performance between open source router firmware and that of closed source router firmware.

In addition to have measured bandwidth, packet loss and response time has also been measured in order to make it easier for companies/people to make informed decisions regarding whether to use open source router firmware or not.

To further help with decision making, a qualitative study was made to gather data regarding how easy each firmware is to configure and how secure they are.

There were some larger differences between the open source router firmware and that of the closed source router firmware. The closed source router firmware performed better when it came to bandwidth, whereas the open source firmware got better response time and overall better stability.

Keywords: Open source, Closed source, Router, Firmware, Performance, Bandwidth, Response time.

Contents

1 Introduction

With open source router firmware being used more and more often in studies and real life, it needs to be properly researched in order for people/companies to be able to make informed decisions regarding if it is something for them or not. Various studies exist for what you can do with open source router firmware, but none of them compares the open source router firmware to the standard router firmware.

An experiment has been carried out in order to be able to help companies and people to make decisions regarding open source firmware using the results of the study. A qualitative study has also been made in order further help decision-making.

2 Background

This chapter contains the explanation of concepts that the reader needs to understand in order to understand the problem definition.

2.1 Closed Source

The opposite of open source is instead software which's source code cannot be modified by anyone but the one who created it. This is called "closed source" since only the original authors are the only ones that may legally modify or copy it. Examples for such hardware is Microsoft word and Adobe Photoshop. Before a user run such a program they must also agree that they will not do anything with the hardware that the authors have not expressly permitted (opensource.com, 2016).

According to Khanjani & Sulaiman (2011), proprietary software is naturally non-competitive due to the important economic aspects between proprietary software and open source that is that open source software is usually cheaper than proprietary software, if not free. They also say that close source restricts interoperability between different manufacturer systems.

One of the mayor advantages of closed source has over open source firmware according to Khanjani & Sulaiman (2011) is the service and support.

2.1.1 Stock router firmware

Stock firmware is the router's operating system in its default form, without any modification made to the code of the firmware. True stock firmware is mainly found in cases where the device and the firmware has the same manufacturer. (addicitvetips.com, 2011)

2.2 Open source

As described by opensource.com (2016), "the term open source refers to something that can be modified and shared because its design is publicly accessible". Meaning that the code that an open source program exist of, can be altered by anyone that has access to the code to suit their needs.

Before the installation of open source, much like with most software, the user must accept the terms of a license. But the terms of an open source license differ much from its counter-part closed source, since the license promote the collaboration and the sharing of the source code, since it also people to make modifications of their own to the source code, which other people then may incorporate in their own projects. Some open source license term also ensures that anyone that modifies or alter the data must then share the source code without charging a licensing fee for it (opensource.com, 2016).

2.2.1 Open source router firmware

Router firmware that has been created, and that are maintained by other people than the ones that created the product.

Response time

Response time is the elapsed time between an input and the service responding to the input. It is the sum of transmission time, service time and wait time. (techtarget.com)

Bandwidth

Bandwidth describes the amount of data that can be transferred between two points, such as between a user and a site. Having higher bandwidth may allow for a faster network. (executionists.com, 2016) According to Ichikawa (2014), bandwidth is an important value to benchmark in a network

2.3 Related works

While similar studies have been made such as the comparison between Open source and closed by Khanjani & Sulaiman (2011), it only covers the theoretical difference between closed and open source as a concept, and doesn't compare any values.

Another study is the study by Palazzi et al. (2010), where they are using open source firmware to create a modification of an access point to shape the transiting network.

The only study found that is within the area of open source firmware is the study by Alm & Björling (2014) on a router using the MIPS architecture, where study and compare the performance difference between open source router firmware, but without the use of a base line in the form of closed source firmware. This study instead covers both the standard closed source firmware of a router, and the open source methods, in order to be able to discuss the difference between the two and help make decision between open and closed source.

3 Problem Definition

This section encompasses the parts that are connected to the problem definition, such as aim, motivation and the research question.

3.1 Aim

The aim of the study is to measure the difference in performance (bandwidth, packet loss, response time), security and ease of configuration between closed source standard router distribution and open source distributions.

3.2 Motivation

While studies show that the use of open source router firmware can achieve both an increase in security and flexibility, while keeping the costs low, they never mention anything about how they perform against the stock firmware in terms of performance. At the time this study was made no other studies did this either, creating a need for such a study to be made.

When choosing whether to use install open source firmware, performance is one of the most important aspects. This study is aiming to reveal if, and how, implementing an open source solution affects the performance of the router. With the help of the results, the study should fill in the gap in scientific knowledge, and should be able to help in making informed decisions regarding if it is worth installing an open source firmware. As measuring bandwidth by itself is not enough of a factor to help making informed decisions regarding performance, packet loss and response time is as well.

While performance is an important aspect of every router, there are many others. Whether to use an open source router firmware can also be affect how easy it is to configure and how secure it is. For this reason, a qualitative study is made in order to be able to compare the two regarding security and ease of configuration. Linux-based routers are when in comparison to commercial routers cheaper and the Linux environment allows for greater flexibly and functionality (Heldenbrand & Carey, 2007). Making open source router firmware a viable replacement, in terms of price and flexibility, to that of buying and using routers using the stock firmware.

A study by Ortega et al. (2009), presents a proof of concept that using open source routers can prevent ARP cache poisoning attacks, ultimately increasing the security of the network. The scheme is also economic, efficient and easy to implement.

3.3 Research question

The research question that will be answered in this study is:

How does the functionality of open source firmware compare against the original closed source firmware of a router?

The research question not only aims to see how much faster/slower one is in comparison to the other, but will also investigate other factors.

3.4 Hypothesis

The hypothesis is that the implementation of open source firmware would increase the overall functionality when a comparison is made to the standard closed-source firmware.

3.5 Objectives

The study consists of a number of objectives. These are done in order to reach the aim of the study. The objectives are the following:

1. Conduct a literature study in order to see gain deeper knowledge of the subject, making it easier to make informed decisions during the study, and to see what studies has been made within the field before.
2. Perform a validity threat analysis in order to find the applicable, and non-applicable validity threats.
3. Construct an experimental design in order to be able to answer the research question of the study.
4. Perform a qualitative study to gather data regarding ease of configuration and the security of the chosen firmware's.
5. Perform the experiment, using benchmarking programs to gather data regarding the bandwidth, packet loss and response time of three different router firmware's.
6. Analyze and interpret the data that was collected during the experiment, with the goal in mind to help people with their decision making regarding open source firmware.

4 Methodology

This chapter contain the methods chosen in order to complete the objectives of the study, alternative methods are also discussed and compared with the chosen method with why it was not chosen in mind. It also covers what validity threats that are relevant for the chosen method.

4.1 Experiment

As stated by Wohlin et al. (2012), experiments are launched when we want control over the situation and want to manipulate behavior directly, precisely and systematically. An experiment, in the form of a benchmark test will therefore suit the study. This allows the collection of large amount of data, making it easy to reveal found patterns.

4.1.1 Alternative methods and discussion

An alternative method may for an experiment in this case has not been found. According to Wohlin et al. (2012), there are three major different types of investigation strategies, a survey, a case study and an experiment. Data collection through the means of a survey is mostly made through the usage of interviews or questionnaires making it unsuitable for benchmarking tests. Whereas a case study, as Wohlin et al. (2012) describes it, is to be used to investigate a single entity and phenomenon. While a case study may have been used instead of an experiment, the results from such a study become harder to interpret and generalize causing an experiment study to be chosen as method in the end.

4.2 Independent variables

As Wohlin (2012) describes it, an independent variable are the values that we can control and change in the experiment. In this study the independent variables are what kind of firmware is running on the router, either if it is open source or closed source.

4.3 Dependent variables

According to Wohlin (2012), the effect of treatments is measured in the dependent variable(s). The dependent variables in this case, end up being response time and bandwidth, and those variables are what the study are investigating by changing the independent variables.

4.4 Experimental design

The experiment measures the performance in form of bandwidth and response in a router running the firmware DD-WRT and Tomato. Another test, using the same methods, but this time for the stock firmware, allows a comparison between the two. The test for each firmware are in an environment where the only difference between each concurrent test is the current firmware of the router. The experiment uses the following components:

- A server computer running Ubuntu 14.04.4 LTS. With Apache version 2.4.7, and running iperf version 3.13.0-86 version.
- A server computer running VMware ESXi with four virtual machines running vyos-1.1.7 acting as routers.
- A client computer running Windows 7 service pack 1 with iperf and hrPING.

- The wireless router Netgear R7000 with stock firmware, which during the experiment is to be changes to Advanced Tomato 1.28 and DD-wrt v24-sp2 (04/16-14 kognac)

Abstract experimental setup
During the experiment the benchmarking program iperf is acts as a load generator in order to generate traffic between the iperf running on the client, and the iperf running on the server. The benchmark is capturing traffic which is later to be analyzed during the analysis.

In order to capture data to analyze the response time, the tool hrPing is running on the client against the server. This does not require that hrPing is running on both computers, as with the iperf program.

4.5 Qualitative study

As the results from the experiment, may not be enough to make a conclusion regarding open source router firmware, a qualitative study is made as well. This information is taken from sources such as the system documentation, man-pages and the official websites for the chosen router firmware.

The following features are examined:

1. Security
2. Ease of configuration

4.6 Method implementation

This chapter contains the motivation for choosing the firmware and benchmarking programs. It also provides an explanation for the variables, and also the plan for gathering data for the result part of the report.

4.6.1 Study of available firmware/software

In order to decide which, open source software, and benchmarking program that is to be chosen for the experiment, a literature study is needed. The study is performed by searching in both casual search engines, such as Google or Bing, but also by searching in scientific search engines such as Google scholar, IEEE, and Springer.

Study of available firmware

While searching for router firmware, it was important to search for open source firmware that was regularly updated and that could work as a potential replacement for the standard firmware. This was needed since a router that is not updated regularly is increasingly a potential target for attacks, as the security decreases over time unless it is updated. It was also important that the firmware had a router that multiple firmware's supported so that the same experiment could be made with the same router (with different firmware), in order to avoid validity faults. It was also taken into account what priorities the firmware had when designed, performance, simplicity etc., in order to secure that the chosen firmware is not fundamentally the same.

A multitude of open source firmware were found and studied, some of them were found unsuitable for the study, such as HyperWRT, due to the support for the project no longer is being supported. The ones found suitable and relevant for the study are the ones mentioned below.

DD-wrt
DD-WRT is a Linux based alternative open source firmware that is suitable for a large variety of WLAN routers and embedded systems. The main emphasis lies in providing easy handling while at the same time support a large number of functionalities within the framework of the respective hardware platform used. It usually is not shipped with the router, meaning that the user needs to install it on top of the stock firmware. (DD-WRT. 2016)

Tomato firmware
Tomato is a small, lean and simple replacement firmware for Broadcom-based routers. It is a Linux based firmware but opt for a better balance between performance and features. It also has a simple interface, making it easy for beginners to use. Much like other open source firmware it needs to be run on top of the stock router firmware. (VPNPICK. 2015)

Study of available benchmarking programs

When searching for a suitable benchmarking programs, certain requirements has to be fulfilled. The requirements are that they need to be able measure response time and bandwidth. In order for the study to be applicable to the real world, it would also be preferable if the benchmarking programs simulated traffic in a way as close to the real thing as possible. The following benchmarking programs were found suitable.

Iperf
A tool used for active measurements of the maximum amount of bandwidth. It also supports the tuning of various parameter such as what protocol should be used, and the buffer. For each test it also reports the bandwidth and packet loss. (Iperf.fr, 2015)

Hrping
A tool that can be used to measure the response time. It works much like the normal ping tool that comes with Windows, but will give the results more accurately. It is also possible tune various parameter such as packet size. (cfos.de, 2015)

4.6.2 Experiment

The experiment aims to measure the difference in a multitude of factors after changing the active firmware of the router responsible for routing traffic in the network. This part contains four parts, two explaining the different experiments that need to be done in order to get the data that the experiment needs, one of the installation process, and the last one shows the network topology.

Installation of firmware

While benchmark tests can start instantly through the standard router firmware, an installation of DD-wrt and Tomato needs to be done router before any tests can be made regarding their overall performance.

Before installation it is important to note that only some router models actually support the installation of open source firmware, therefore it is important to go to the open source

firmware website and look at the devices it supports. As there is different firmware for each individual router model, it is important to look for a firmware matching the router.

The first step to installing open source firmware is to go to the site for the firmware in question, e.g. Tomato firmware homepage, and find the version suitable for the router and download it.

The second step is the actual installation. This is done by firstly accessing the router through the web interface by entering the IP of the gateway in a web browser. And then go into the "firmware" section of the router and choose "firmware upgrade", and choose the file with the filename ".bin". After pressing upgrade, the installation begins.

In this study, the installation process needs to be done twice, once for DD-wrt and one for Tomato.

Network topology

In order to test the router in a more realistic environment, a bigger network is setup, as there usually exist more than one router in a network infrastructure. By implementing other routers it should also be possible to see how the router cooperates with other routers to complete the task at hand. This topology can be seen below in figure 1. The topology allows packets to flow both via vyos2 and vyo3 to each the server/client. As no hop-distance has been setup in the vyos routers, the packages are often split 50/50 and sent across both vyos2 and vyos3 to reach the destination.

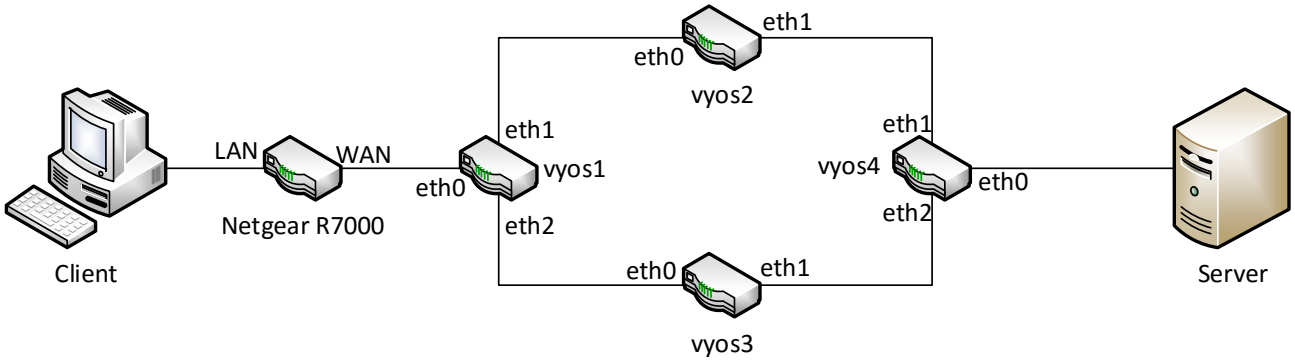


Figure 1 – Network Topology

To see more detailed information about the setup of the network, see Appendix B.

Benchmark Bandwidth

Iperf gathers data regarding bandwidth. Iperf is runs on both the server and the client in the experiment, and is generating TCP traffic between the two which Iperf is also gathering.

For iperf to work correctly one of the sides needs to have iperf in server mode, this is done on the server through the use of the command:

```
iperf3 -s
```

For the client side to start benchmarking against the server the following command has to be run:

```
iperf3.exe -c 10.0.0.2 -4 -t 60
```

This starts the benchmark against the IP-address of the server, with the parameter -4 causing all the traffic to be IPv4 traffic, and the parameter "-t 60" to make the test run for 60 seconds giving a value for each second that passes.

This test was then run five times. Running the tests five times, with each test giving 60 results, allows the gathering of 300 values. To further increase validity, there is a wait time of 5 minutes between the tests, to make sure that the tests do not affect each other.

Benchmark packet loss

For benchmarking packet loss iperf is running, but this time using the UDP protocol instead of the standard TCP. Much like when bandwidth was benchmarked the server is put into server mode through the use of the command:

```
iperf3 –s
```

In order for the client to start benchmarking against the server, the following command has to be run:

```
iperf3.exe -u -l 32k -b8m -c 10.0.10.2 -4 -t 60
```

The parameter –u causes iperf to start the benchmarking using UDP packets, and the parameter –l causes the window size of the packets to become 32kb. As the standard bandwidth when benchmarking UDP is set to a 1MB limit, the parameter –b is changes the bandwidth limit. The size for this parameter is changes five times during the experiment in order in order to see how it affect packet loss. Lastly the –t parameter makes it so that the test runs for one minute.

Benchmark Response time

This benchmark uses the tool hrping to gather data regarding the response time. The benchmark is using the client to send pings toward the server in order to gather data regarding the speed in which the server responds. In order to test the size of a packet affects the speed of which the server would respond, the test is run five times with five different packet sizes. This is achieved by running the following command:

```
hrping.exe 10.0.10.2 -l 32 -n 50
```

This sent an echo request using the ICMP protocol to the server, to which the server is answering by sending an echo reply back to the client. The parameter "l" decides that the packet is 32 bytes long. By modifying the parameter "l" it is possible to measure how the size of a packet affects the time in which it takes for the server to respond. This size is changes 5 times during the test to measure the differences in response time in relation to the size of the packet. The parameter 50 causes test to consist of 50 pings, so that there are enough values to draw conclusions from. (cfos.de)

After a firmware has run through the tests, another firmware is installed on the router instead in order to gather data on the next one.

4.6 Validity

This section handles the validity threats that may be applicable to this study, and describes the four categories of which the threats come from.

4.6.1 Validity threat categories

The four validity threat categories according to Wohlin et al. (2012) are:

Internal Validity
"Threats to internal validity are influences that can affect the independent variable with respect to causality, without the researcher's knowledge." Wohlin et al. (2012).

Internal validity is about the credibility of a study, such as if the right measuring tool has been used at the right time. Since the experiment is done in the local area network, the risk of changes in the environment that may affect the results is relatively low, but to decrease this risk further, the measurements are taken place one after another at the same occasion.

External Validity
"Threats to external validity are conditions that limit our ability to generalize the results of our experiment to industrial practice" Wohlin et al. (2012).

To decrease the risk of eventual external threats, each benchmark runs multiple times to gather large amount of data, since each additional measurement increases the validity. It also makes it easier to discover patterns from the data.

Construct Validity
"Construct validity concerns generalizing the result of the experiment to the concept or theory behind the experiment" Wohlin et al. (2012).

Construct validity is the relationship between the construction of an experiment and the result. Validity threats that fall within the construct validity category are threats that affect the result through the means of the studies approach and design choices. It is therefore important to carefully plan and evaluate the experiment before performing it.

Conclusion Validity
"Threats to the conclusion validity are concerned with issues that affect the ability to draw the correct conclusion about relations between the treatment and the outcome of an experiment" Wohlin et al. (2012).

4.6.3 Validity threats

Each of the four threat categories contains many threats, some of the most important ones to this study are mentioned below. A complete list can be found in Appendix C. Further discussion about the validity threats takes place after the results in study.

Fishing and error rate, is very relevant to the study, but by throwing away all gathered data if a method change occurs during the experiment it is handled. It is also prevented by presenting data all data.

Low statistical power, is avoided through the usage of large datasets, allowing the conclusion to be drawn from a large amount of data.

Violated assumptions of statistical tests, through the usage of histograms, a statistical test, it becomes easy to see if the data follow a normal distribution.

4.7 Ethics

Since the study is an experiment and a literature study, many ethics problems that usually occur when it comes to case studies and surveys are left out.

5 Results

In this chapter the results gotten from the benchmarks is presented. Each test is presented with a short explanation.

5.1 Bandwidth

This section consists of two parts, one of which presents the results for the TCP bandwidth benchmark, and one that presents the results of the UDP benchmark.

5.1.1 TCP

In table 1 the calculated mean value for the bandwidth of each firmware is shown. The deviation for each firmware is also shown. The bandwidth is presented in Mbit per second. The higher the value the better, as that means that the traffic is moving faster. A lower deviation value is better.

Table 1 - Bandwidth TCP

Bandwidth in Mbit/s			
	Stock	DD-wrt	Tomato
Mean	571,8387097	291,1774194	303,1451613
Deviation	63,69791001	13,68381298	9,415003475

In figure 2, a box plot is shown for the bandwidth benchmarks of the firmware. The y-axis shows the speed in Mbit/s.

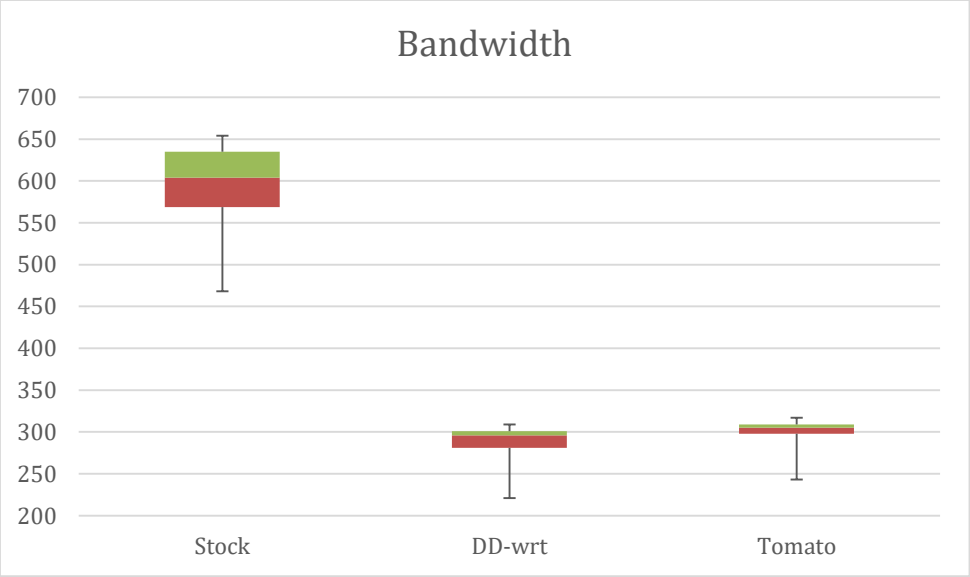


Figure 2 - Bandwidth line chart

In figure 2 the stock firmware got higher bandwidth with open source firmware's performing at almost half the speed. Worth noting is that the two open source firmware's are much more stable than the stock firmware.

5.2 Packet loss

Table 3 shows the packet loss for the different amounts of bandwidth limits for when the UDP protocol was used. It can be seen that the pack loss was kept at a low rate for the first four tests without much of a difference.

Table 2 – Packet loss

	Packet loss in percent		
Bandwidth in MB	Stock	DD-wrt	Tomato
8	2,5%	0,66%	4,2%
16	2,3%	1,3%	3,8%
32	2,1%	1%	2,5%
64	1,6%	0,64%	1,7%
128	53%	11%	1,2%

Table 2 is not be presented in graphical form as the test run with iperf only gave the value of the packet loss in percent for each test.

5.3 Response time

In Table 3, the results of the response time benchmark, hrping, is shown. The response time is presented in ms (miliseconds. Each row presents the results from the benchmark of the certain packet length, e.g. 32b, and the columns show which firmware that was being benchmarked. For each benchmark the mean value for the 50 tests are shown, together with the calculated deviation. A lower mean value is better as that means the client got a response faster, a lower value being better is true for the deviation as well since it means that the connection was stable and didn't peak much.

Table 3

Packet length (in b)	**Response time (in ms)**					
	Stock firmware		DD-wrt		Tomato	
	Mean	Deviation	Mean	Deviation	Mean	Deviation
32	1,4442	0,117173863	1,33772	0,143038462	1,33122	0,132350384
64	1,44718	0,180307073	1,383	0,604368822	1,3772	0,150839014
128	1,53162	0,155837944	1,33988	0,153448208	1,46944	0,13931643
256	1,50886	0,113378004	1,43042	0,163727136	1,48104	0,10941756
512	1,49676	0,14217732	1,32414	0,17558776	1,46736	0,14909903

From Table 3 it is possible to see that Tomato, with DD-wrt shortly after, was the best at handling packets with the packet length of 32b and that the stock firmware was the worst. Worth noting is that the stock firmware deviated the least, when compared to the open source firmware.

When the packet length grew, DD-wrt instead became the one with the lowest mean response time, with Tomato second as can be seen in the test results for the 512b test.

5.3.1 Box plots

In this chapter the box plots for the packet size of 32b and 512b for each firmware is presented. As the results didn't differ much between each jump, e.g. 32 to 64, these is not presented in this section, though they are all available in appendix.

32b packet length

The results for the 32b packet length benchmark have been compiled into a box plot as seen in figure 3. The combined box plot contains the results for all the benchmarks for the packet length of 32b for all three firmware's. In the figure the y-axis for the box plot is shown in ms.

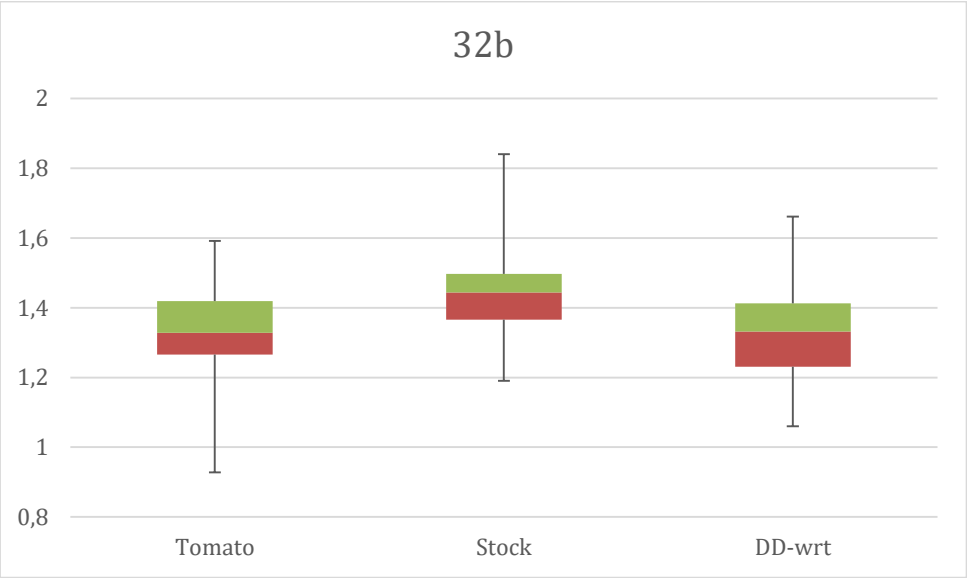


Figure 3 Box plot for 32b packet length

In figure 3 it is possible to see how spread out the values were during the test for 32b packet length.

512b packet length

In figure 4, the frequency for the packet length of 512b for each firmware are presented.

Figure 4 – Box plot for 512b packet length

In figure 4, the box plot for pings with the packet length 512b is shown. When compared to when 32b was used, the values now deviate more.

5.4 Qualitative study

This chapter contains the results of the qualitative study.

5.4.1 Security

The security features found on the stock firmware was also found on the open source router firmware's. This includes such features as WPA2 encryption, and a working firewall. While all investigated features on the stock firmware were found on the open source firmware, it was not the same the other way around. The open source router firmware allowed many other security features to be configured, such as the ability to mail the user if an emergency happens, or monitoring certain devices in the network to discover if they are suspicious.

One function which was turned on by default on the closed source firmware, but not on the open source firmware, was the function WPS. This is bad as WPS has been found to be vulnerable to brute-force attacks. While it could be disabled by configuring the router, it may be hard for unexperienced users to know that WPS contains vulnerabilities and how to disable it.

An advantage found with the open source firmware's was the ability to be able to turn off most functions, this was limited to only a few functions on the closed source firmware. The reason that this is advantageous is that you can turn of a function if it turns out that the function contains security flaws that may compromise network.

The open source router firmware's allowed access to the source code, which for example allows more experienced users to modify the code in cases that security flaws are found, e.g. backdoors. This was not possible on the closed source firmware, forcing the user to buy a new router to be secure unless the manufacturer releases an official patch which solves the issue.

5.4.2 Ease of configuration

Which firmware which is easier to configure depends on what is going to be configured. While making simpler changes to the router was the easiest to do with stock firmware, more complicated changes to the stock firmware required that the user knew where to look and was comfortable with the firmware. This was not true for the open source firmware, where the difficulty of configuring the router was about the same from the beginning.

6 Analysis

This chapter contains the analysis for the results presented in the previous chapter.

6.1 Bandwidth

The bandwidth for the routers presented in Table 1 showed that the mean bandwidth of that of the stock firmware was far superior over that of the open source firmware's, having almost twice the amount of bandwidth, making it the clear winner in terms of pure bandwidth. Whereas Tomato performed better than DD-wrt in terms of bandwidth, they didn't differ much.

Looking at the deviation shown in Table 1, it is possible to see that while the stock firmware had the best bandwidth, is also had a much higher deviation than the open source firmware's. This is further seen in figure 2, the deviation was far greater when looking at the stock firmware compared to that of the open source ones.

Making the stock firmware a clear winner in terms of bandwidth when compared to the open source firmware's.

6.2 Packet loss

The packet loss shown in Table 2, shows that during the first four test not much is happening in terms of packet loss. During the test for the 128mb bandwidth limit the packet loss instead increased to a much larger amount for the stock and DD-wrt firmware. Where the increase was much larger for the stock firmware (increased from 1,6% to 53%) the DD-wrt still suffered a pretty large increase (increased from 0,64% to 11%). Whereas the stock and dd-wrt firmware increased the amount of dropped packets, the tomato firmware instead got a lower result than it had for the test of the 64mb bandwidth limit.

Making the tomato firmware a clear winner in terms of packet loss.

6.3 Response time

Looking at the response time presented in Table 3, there is normally an increase in the mean response time as the packet length increases. This is true for the deviation for each of the firmware's as well. Worth noting is the mean response time for the DD-wrt firmware, as the response time wasn't affected much by the increase in packet length, even having a lower response time for the packet length of 512b than that of 32b.

Though the mean response time for the DD-wrt firmware was the best out of the three is also had the highest deviation on four out of five packet lengths, whereas the stock firmware had the lowest deviation in four out of five packet lengths. Both of the open source firmware's on the other hand had a greater spread of the values.

Making the open source firmware's the winners regarding the response time, whereas the stock firmware had more stable results with less deviation.

7 Conclusion

The aim of the study was to performance in terms of bandwidth, packet loss, and response time with the goal in mind to help people making informed decisions regarding open source router firmware. This was to be made by benchmarking the stock firmware and two open source firmware's and then compare the results.

The results showed that there were some differences between the open source firmware and the stock firmware, with the main difference being the bandwidth where the stock firmware at times performed at twice the speed when compared to the open source firmware. But when comparing the stability, in terms of deviation, the open source firmware proved to be more efficient than that of the stock firmware.

Another big difference was the amount of lost packets when it came to using large amounts of UDP bandwidth. The open source firmware performed over all better when it came to packet loss having a large advantage over that of the stock firmware.

The Open source router firmware was also better when it came to response time, but only with a slight margin. Instead the stock firmware had a lower deviation when it came to response time.

When it comes to decide whether or not for a user to install open source router firmware, it comes to two factors. One is whether or not it is important to have high bandwidth, and if the owner of the router has knowledge in configuring one. If it is important for a user to have high bandwidth, while having little knowledge of configuring routers it would be recommended for them to stick to the stock firmware as increasing bandwidth of an open source router firmware to that of the stock firmware would require the user to do more advanced configurations. If the user on the other hand is more knowledgeable it is instead recommended to switch to an open source router firmware as it is easier to perform more complex configurations and more features becomes available.

If it instead is a company where it may be more important to have more stable networks without peaks in the traffic and high response time, an open source firmware is recommended. The reason for companies to choose open source router firmware is further strengthened by the fact that they get access to more security features which can be used to increase the network security of the company.

8 Discussion

The study was an experiment. While the experiment delivered results regarding the overall performance of stock firmware vs open firmware from which conclusions could be drawn, it feels like it would have been good to have performed a theoretical study for the firmware's other features as well. This because there might be features exclusive to the open source firmware's that can boost some factors such as the bandwidth. But as there was no time near the end it was decided to be left unhandled.

Where a lot of values was gathered for the experiment it feels like an even larger amount could be collected, like instead of running one of the tests for 60 seconds it could be run for a couple of hours in order to see if the span of time would affect the results, it would also be easier draw conclusions. But as this was something that was first thought of during the analysis it was left out.

Worth noting is that the performance of the router may vary due to hardware differences. This can be differences such what computer is used, meaning that using another computer than what was used in this study may give other results. This is also true for the router which was used, meaning that running open source router firmware on another router than the R7000 that was used in the study may give varying results.

The results of the study very pretty straightforward, meaning that it was easy to draw conclusions from the data.

Overall the objectives of the study were reached in such a way that was found satisfactory, meaning that the study answered the question that it asked while keeping the validity at a good level.

8.1 Future studies

There are still many things that need to be studied regarding open source router firmware, such as their built-in-functions and how they may affect the performance.

This could be done by having an open source router firmware, benchmark it, and then start activating various functions that may affect the performance. After each change in the configuration the benchmark could be ran again to see how the effect on the performance. By doing such a study this study might be complemented since this study didn't take built it functions into account and only measured the open source router firmware as it was.

9 References

Alm, A. and Björling, J. (2016). *Prestandautvärdering av firmwares baserade på öppen källkod för routrar/brandväggar på MIPSarkitektur*. [online] Available at: http://www.diva-portal.org/smash/get/diva2:733318/FULLTEXT01.pdf [Accessed 1 Mar. 2016].

Bradner, S. (1991). Benchmarking terminology for network interconnection devices [online] Available at: http://www.ietf.org/rfc/rf1242.txt [Accessed 1 Mar. 2016].

Cfos.de. (2016). *Ping Utility hrPING v5.06 - hrPing - High-precision ping utility - cFos Software*. [online] Available at: https://www.cfos.de/en/ping/ping.htm [Accessed 24 May 2016].

Christensson, P. (2016). *Firmware Definition*. [online] Techterms.com. Available at: http://techterms.com/definition/firmware [Accessed 1 Mar. 2016].

Dd-wrt.com, (2016). *About DD-WRT | www.dd-wrt.com*. [online] Available at: http://www.dd-wrt.com/site/content/about [Accessed 1 Mar. 2016].

Executionists | Web Design, Development and Marketing Agency. (2013). *An Explanation Of Bandwidth: What It Means And How Much You Need - Executionists | Web Design, Development and Marketing Agency*. [online] Available at: http://executionists.com/an-explanation-of-bandwidth/ [Accessed 24 May 2016].

GUEANT, V. (2016). *iPerf - The TCP, UDP and SCTP network bandwidth measurement tool*. [online] Iperf.fr. Available at: https://iperf.fr/ [Accessed 24 May 2016].

Khanjani, A. and Sulaiman, R. (2011). The aspects of choosing open source versus closed source. *2011 IEEE Symposium on Computers & Informatics*

NETGEAR. (2016). *AC1900 - Nighthawk AC1900 Dual Band WiFi Router*. [online] Available at: http://www.netgear.com/home/products/networking/wifi-routers/R7000.aspx?cid=gwmng [Accessed 24 May 2016].

The linux router: an inexpensive alternative to commercial routers in the lab. (2007). *Journal of Computing Sciences in Colleges*, Volume:23(1), p.127.ISSN:

Opensource.com, (2016). *What is open source?*. [online] Available at: https://opensource.com/resources/what-open-source [Accessed 1 Mar. 2016].

Open-source.gbdirect.co.uk, (2016). *Benefits of Using Open Source Software*. [online] Available at: http://open-source.gbdirect.co.uk/migration/benefit.html#flexibilityfreedom [Accessed 1 Mar. 2016].

Openwrt.org, (2016). *OpenWrt*. [online] Available at: https://openwrt.org/ [Accessed 1 Mar. 2016].

Ortega, A., Marcos, X., Chiang, L. and Abad, C. (2009). Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt. *2009 Latin American Network Operations and Management Symposium*. DOI: 10.1109/LANOMS.2009.5338799

Palazzi, C., Brunati, M. and Roccetti, M. (2010). An OpenWRT solution for future wireless homes. *2010 IEEE International Conference on Multimedia and Expo*.

SearchNetworking. (2016). *What is response time? - Definition from WhatIs.com*. [online] Available at: http://searchnetworking.techtarget.com/definition/response-time [Accessed 24 May 2016].

Softpedia, (2016). *Download Web-bench 1.5 for Linux*. [online] Available at: http://linux.softpedia.com/get/System/Benchmarks/Web-bench-1378.shtml [Accessed 1 Mar. 2016].

VPN Service Reviews 2016 | VPNPick.com, (2014). *DD-WRT vs. Tomato vs. Open WRT? - VPN Service Reviews 2016 | VPNPick.com*. [online] Available at: http://vpnpick.com/dd-wrt-vs-tomato-vs-open-wrt/ [Accessed 1 Mar. 2016].

Wohlin, C. (2012). *Experimentation in software engineering*. Berlin: Springer.

Appendix A – Virtual ESXi

Processor:	Intel Core i7 920 (2,67Ghz)
Memory (RAM)	16GB
Harddrive:	160GB
Version:	6.0

Virtual VyOS	
Operatingsystem	: 1.1.7
Processor:	1 CPU, 1 Core
RAM:	1GB
Harddrive:	16GB

Appendix B – Network topology

Name	Interface	IP-address	Subnet
Client	LAN	10.0.0.2	255.255.255.0
Netgear R7000	LAN	10.0.0.1	255.255.255.0
Netgear R7000	WAN	192.168.30.1	255.255.255.0
VYOS1	Eth0	192.168.30.2	255.255.255.0
VYOS1	Eth1	192.168.10.1	255.255.255.0
VYOS1	Eth2	192.168.20.1	255.255.255.0
Vyos2	Eth0	192.168.10.2	255.255.255.0
Vyos2	Eth1	172.168.10.1	255.255.255.0
Vyos3	Eth0	192.168.20.2	255.255.255.0
Vyos3	Eth1	172.168.0.1	255.255.255.0
Vyos4	Eth0	10.0.10.1	255.255.255.0
Vyos4	Eth1	172.168.10.2	255.255.255.0
Vyos4	Eth2	172.168.0.2	255.255.255.0
Server	LAN	10.0.10.2	255.255.255.0

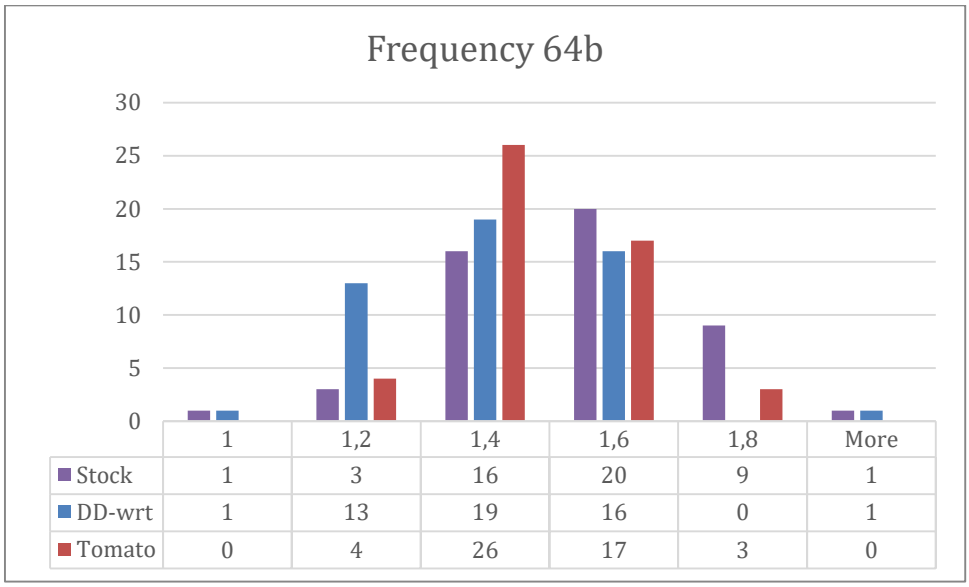
Appendix C - Validity threats

Validity threats	Applicable	Preventable	Handled
Low statistical power	Yes	Yes	Handled by collecting large quantities of data during the experiment
Violated assumption of tests	Yes	Yes	By choosing to use a statistical test that that makes it easy to see if the data follow a normal distribution.
Fishing and error rate	Yes	Yes	If a method changes occur, all collected data will be thrown away. All collected data for the chosen method will be presented.
Reliability of measures	Yes	Yes	By performing the test at the same time, on the same day concurrently.
Reliability of treatment implementation	No	-	Not applicable.
Random irrelevancies in experimental setting	Yes	Yes	Is handled by having a virtual network which is not connected to the internet, leaving only small unpreventable threats such as power outage.
Random heterogeneity of subjects	Yes	Yes	Since all test will be performed on the same hardware, it is handled.
History	Yes	yes	Applying the treatments at different times won't affect the end result, as the computers isn't connected to the internet they won't update unless found necessary.
Maturation	No	-	A computer won't act differently as time passes.
Testing	No	-	Computers can't learn how the test is conducted.
Instrumentation	Yes	Yes	Handled by reading the available documentation for each of the programs that will be used for benchmarking, but also the documentation for each firmware.
Statistical regression	No	-	Not applicable on the study.
Selection	No	-	The subject is a computer. No humans.
Mortality	No	-	The subject is a computer. No humans.

Ambiguity about direction of casual influence	Yes	Yes	Handled through the use of randomization to determine independent variables.
Interaction with selection	No	-	Not applicable on the study.
Diffusion or imitation of treatments	No	-	Not applicable on the study.
Compensatory equalization of treatments	No	-	Not applicable on the study.
Compensatory rivalry	No	-	Not applicable on the study.
Resentful demoralization	No	-	Not applicable on the study.
Inadequate preoperational explication of constructs	Yes	Yes	Handled through the usage of scientific papers.
Mono-operation bias	Yes	No	The study consist of only an experiment. Could have been handled by having a theoretical study as well.
Confounding constructs and levels of constructs	Yes	Yes	Handled by using two methods. An experiment and literature study.
Interaction of different treatments	No	-	Not applicable on the study.
Interaction of testing and treatment	No	-	Not applicable on the study.
Restricted generalizability across constructs			Not relevant
Hypothesis guessing	No	-	Not applicable on the study.
Evaluation apprehension	No	-	Not applicable on the study.
Experimenter expectancies	No	-	Not applicable on the study.
Interaction of selection and treatment	No	No	Not applicable on the study.

Interaction of setting and treatment	No	-	Not applicable on the study.
Interaction of history and treatment	No	-	History will not have a large effect on the outcome.

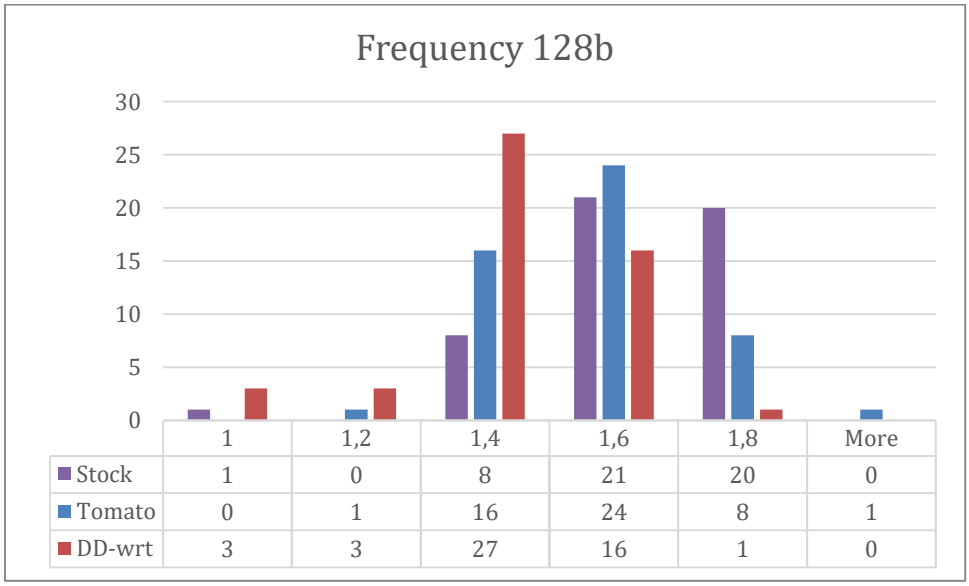
Appendix D Histograms



	1	1,2	1,4	1,6	1,8	More
Stock	1	3	16	20	9	1
DD-wrt	1	13	19	16	0	1
Tomato	0	4	26	17	3	0

Figure 1

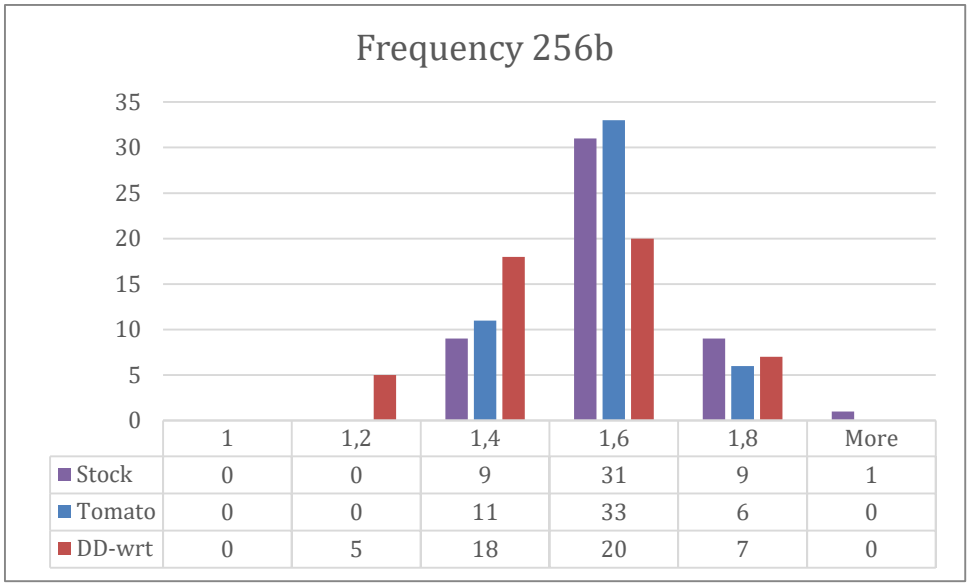
Figure Contains the histogram for 64b packets.



	1	1,2	1,4	1,6	1,8	More
Stock	1	0	8	21	20	0
Tomato	0	1	16	24	8	1
DD-wrt	3	3	27	16	1	0

Figure 2

Figure 2 contains the histogram for 128b packets.



	1	1,2	1,4	1,6	1,8	More
Stock	0	0	9	31	9	1
Tomato	0	0	11	33	6	0
DD-wrt	0	5	18	20	7	0

Figure 3

Figure 3 contains the histogram for 256b packets.

Appendix E Popular Summary

Every router a person or a company buy comes with an already installed software, but for some routers it is possible to install other software to take its place. This study aims to compare the installed software that came with the router, to that of a software installed by the user to take its place.

One of the factors that are used for the comparison is the performance which in this case includes how fast data can be sent through it, how much data is dropped during a transfer, and how fast a device responds.

In addition to performance, a comparison is made which compares the two in terms of how easy each one is to configure, and how secure they are.

The results of the study can help people and companies to make informed decisions regarding if they want to stay with the original software or switch it to another one.

The study found that there are some major differences between the two, this is true in terms of performance, and how secure and how easy they were to configure. While the original software was faster, the replacement software was better at other things and was overall more stable.